**How to monitor Jenkins metrics using Prometheus & Grafana?**

**What is Prometheus?**Prometheus is an open source monitoring tool mainly used for metrics monitoring, event monitoring, alert configuration, etc. Prometheus is designed to monitor targets, server, databases, standalone virtual machines etc. Prometheus uses a powerful query language called **“PromQL”.**



Prometheus Logo

**Prometheus configuration file and components:**

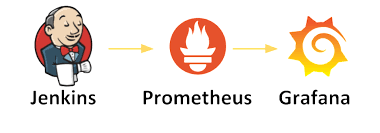
* **prometheus.yml:-**It is the configuration file for Prometheus where we can do all the changes regarding the configuration of Prometheus.
* **Promtool:-**It is the command-line utility tool used to verify the configuration of Prometheus.
* **PromQL:-**It is the strong query language used by Prometheus.

**What is Grafana?**Grafana is free and open source visualization tool that provides various dashboards, charts, graphs, alerts for the particular data source. Grafana allows us to query, visualize, explore metrics and set alerts for the data sources. We can also create our own dynamic dashboard for visualization and monitoring. We can save the dashboards and share it with other members also. We can also import external dashboards.



Grafana Logo

In this tutorial we are going to monitor the various **Jenkins** metrics using **Prometheus** and **Grafana**& will create dashboard in Grafana



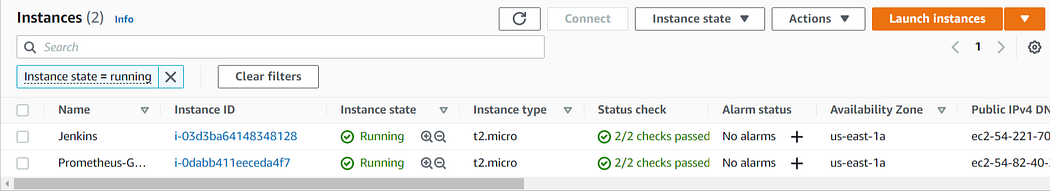
Jenkins + Prometheus + Grafana

**Prerequisite:-**

* Basic overview of **Prometheus** & **Grafana**
* AWS account
* A server with **Jenkins** installed on it

**Step 1:- Create a server**

* In this tutorial I have created the server of **Ubuntu 20.04**AMI
* Create **Ubuntu 20.04**server in **AWS**. In this server we will install **Prometheus & Grafana**
* For **Prometheus & Grafana** server we need to open **9090** & **3000** port & port **22** for SSH



Servers

**Step 2:- Install Prometheus**

* Download and install Prometheus on the server from this [**link**.](https://prometheus.io/download/#node_exporter)

https://prometheus.io/download/#node\_exporter

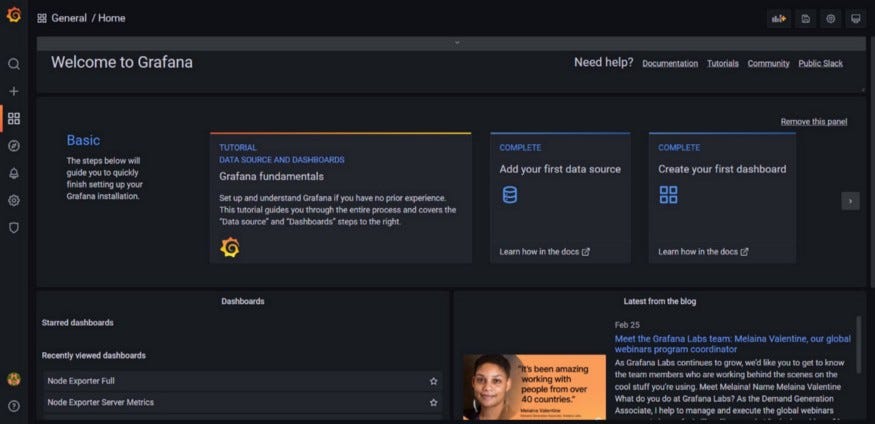
* In order to start the Prometheus we need to run ./prometheus in the Prometheus directory
* We can view UI of **Prometheus** at **<Public IP:9090>**

**Step 3:- Install Grafana**

* Download and install Grafana on the same server on which we have installed Prometheus from this [**link**](https://prometheus.io/download/#node_exporter)

https://prometheus.io/download/#node\_exporter

* In order to start the Grafana we need to run ./bin/grafana-server command from the Grafana directory
* When you open UI of Grafana for the first time it will ask for Username and Password. The default Username & Password is **admin.**You can change the password afterwards
* We can view UI of Grafana at **<Public IP: 3000>**



Grafana UI

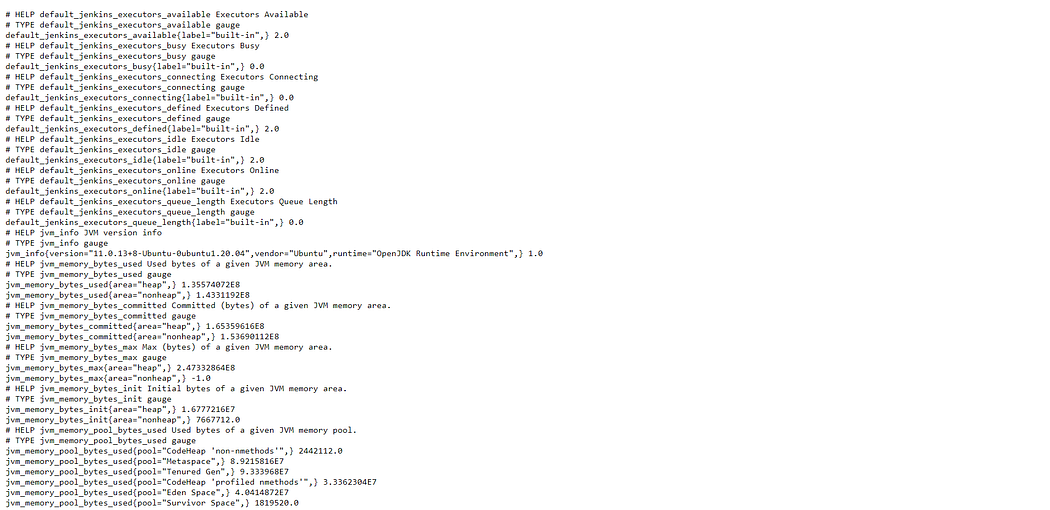
**Step 4:- Install Prometheus plugin in Jenkins**

* We need to install the plugin of **Prometheus** in Jenkins so that Prometheus can gather all the metrics of Jenkins
* In Jenkins click on manage plugin and search for **Prometheus metrics plugin** and click on install



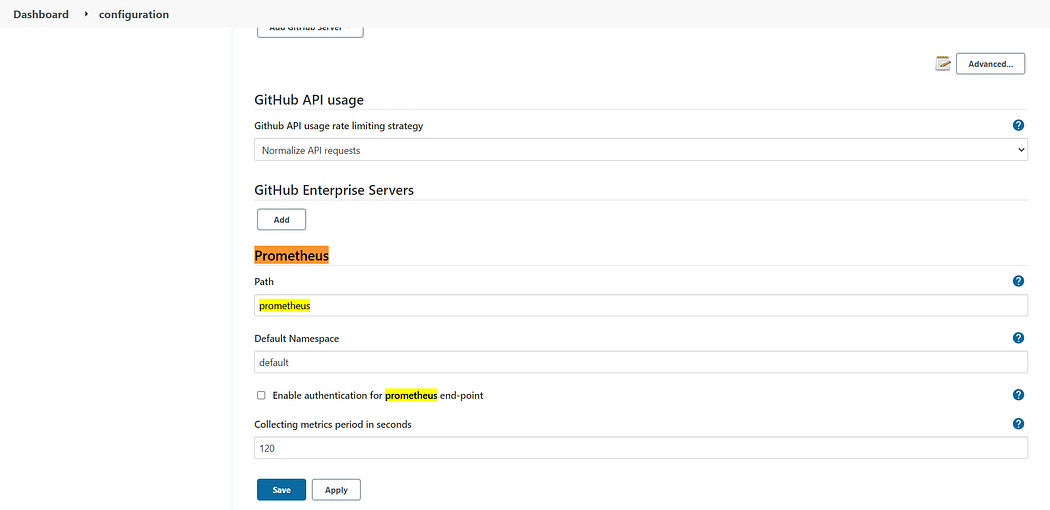
Prometheus Metrics Plugin

* The default path for Jenkins metrics is **<Public-IP:8080/prometheus>**



Jenkins Metrics

* We can also change the default path of Jenkins metrics. Click on **Manage Jenkins -> Configure System -> Search for Prometheus**and change the path.



Default Path

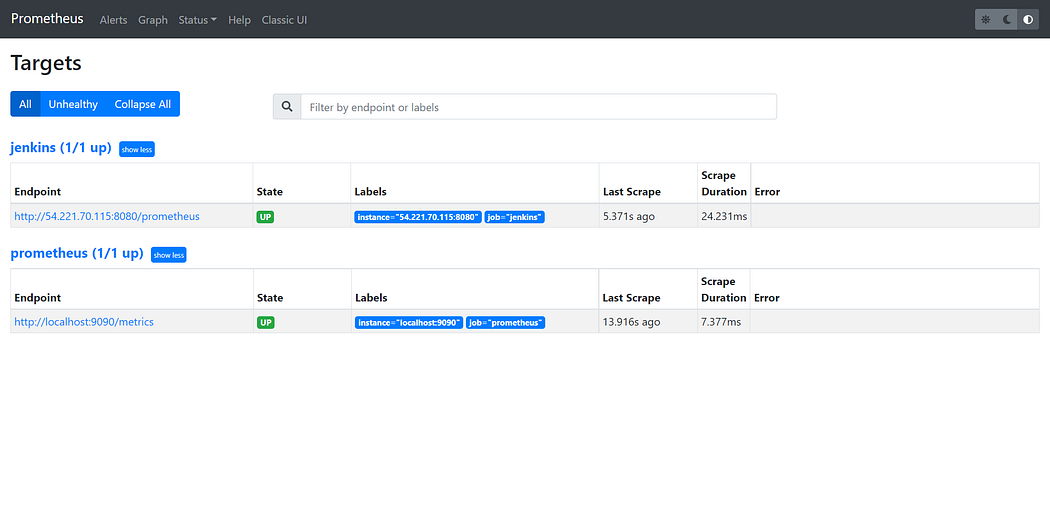
**Step 5:- Modify the configuration file of Prometheus**

* Prometheus manages the configuration file named **prometheus.yml**in which we can define various configuration like**alerts, scrape\_configs** etc.
* In order for Prometheus to gather the metrics of the Docker node we need to define below code in **prometheus.yml**under the **scrape\_configs**stanza

- job\_name: "Jenkins Job"

metrics\_path: /prometheus  
 static\_configs:  
 - targets: ["<Public IP of Jenkins Node:8080"]

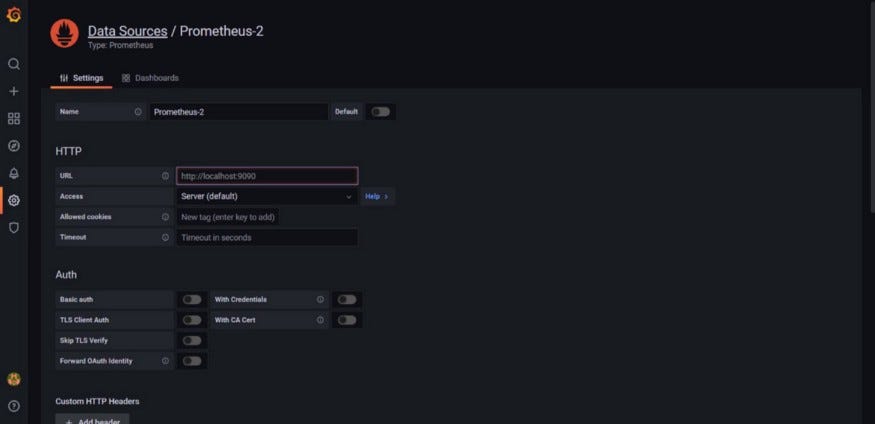
* After adding the code we can check our node as a target in the Prometheus. It will take some time for Prometheus to gather all the metrics and display the state of node to **Up**.



Targets

**Step 6:- Add data source in Grafana**

* Now in Grafana we need to add Prometheus as the data source
* After that, in the URL section of we need to enter the IP address of the Prometheus server and **9090** port



Data Source

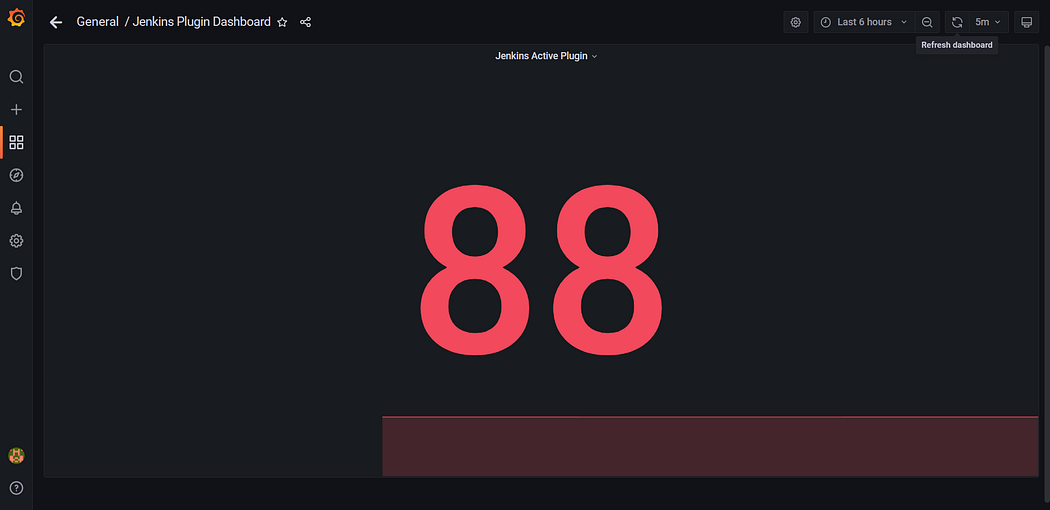
**Step 7:- Create a Dashboard**

* In Grafana we can create various kinds of dashboards as per our need
* So, for this tutorial we will create a dashboard with 4 panels
* First panel will display **No. of active plugins,**Second panel will display **No. of Plugins which required update,**Third panel will display **No. of inactive plugins,**The last panel will display **No. of failed plugins.**
* Click on **create**-> **Add Panel**
* In the Metrics browser section add below query. The below query will fetch the number of **active plugins**

jenkins\_plugins\_active{}

jenkins\_job\_count\_value

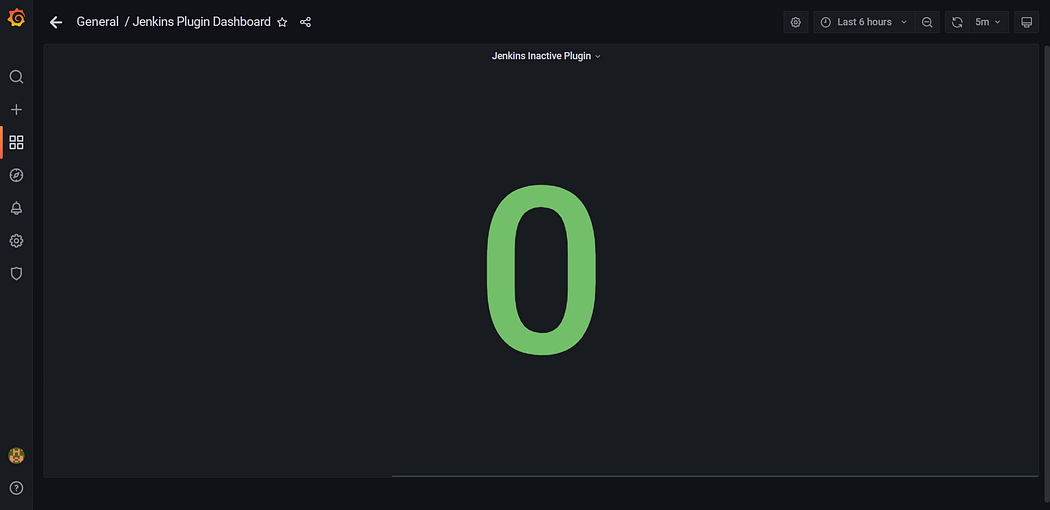
default\_jenkins\_build\_last\_build\_duration\_milliseconds



Active Plugins

* You can change the visualization to **stat**
* Click on **Add Panel**to add second panel
* In the Metrics browser section add below query. The below query will fetch the number of**inactive plugins**

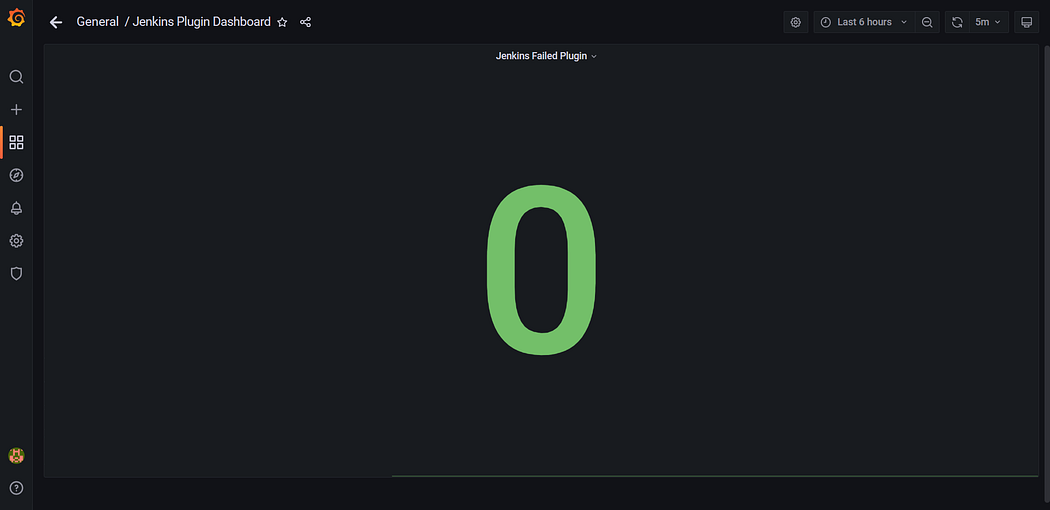
jenkins\_plugins\_inactive{}



Inactive Plugins

* Click on **Add Panel**to add third panel
* In the Metrics browser section add below query. The below query will fetch the number of**failed plugins**

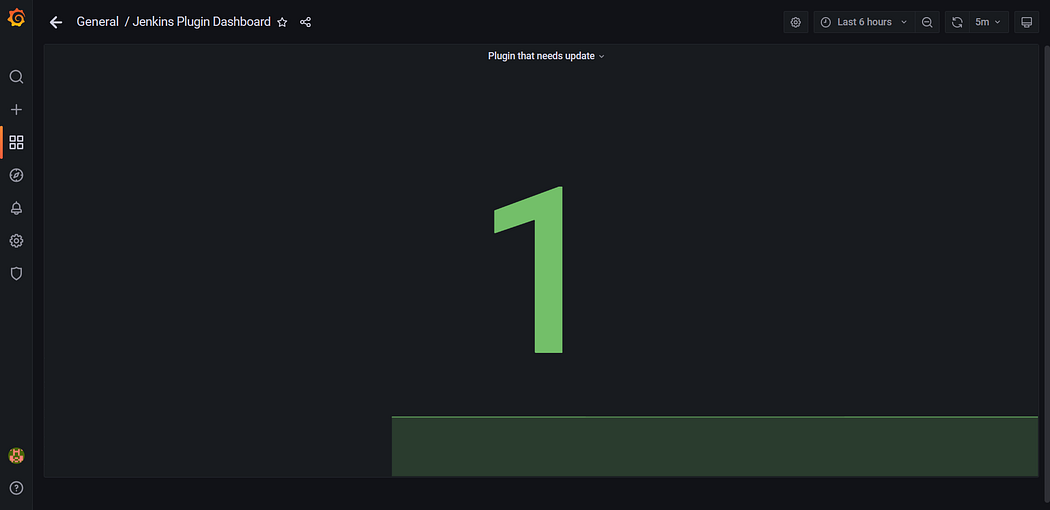
jenkins\_plugins\_failed{}



Failed Plugins

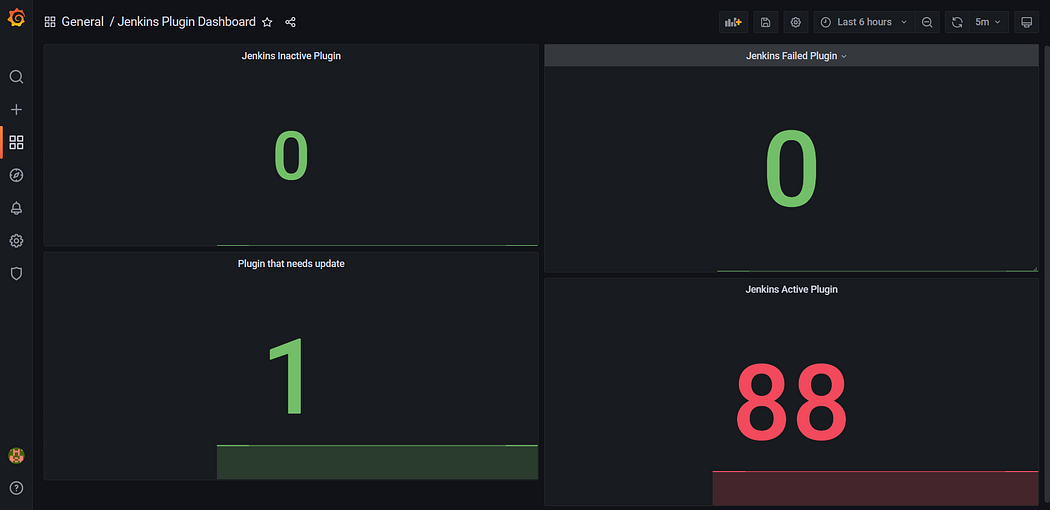
* In the Metrics browser section add below query. The below query will fetch the number of**plugins which required the update**

jenkins\_plugins\_withUpdate{}



Update required Plugins

* Click on save dashboard button to save the dashboard & give it a name of your choice
* The end result of a dashboard will look like:



Dashboard

* You can change the time of frequency after which a dashboard should load automatically. In my case, I have set it to 5m.